Author Query Form

Journal: AURO **Article ID:** 9127	**Please send your responses together with your list of corrections via web (preferred), or send the completed form and your marked proof to:** Akademijos 4, LT-08412 Vilnius, Lithuania fax: +370 5 2784 091 e-mail: vtexspr-corrections@vtex.lt

Dear Author,

During the preparation of your manuscript for typesetting, some questions have arisen. These are listed below.

Queries and/or remarks

Location in article (line)	Query / remark	Response
2/140,336,946,1357,1402, 1418	Please provide the missing reference	
15/1550	Please cite or remove the uncited Refs. marked with <unc>.	

Many thanks for your assistance

Metadata of the article that will be visualized in Online First

Journal Name	Autonomous Robots	
Article Title	A biomimetic, force-field based computational model for motion planning and bimanual coordination in humanoid robots	
Copyright holder	Springer Science+Business Media, LLC This will be the copyright line in the final PDF.	
Corresponding Author	Family name	Mohan
	Particle	
	Given Name	V.
	Suffix	
	Division	Robotics, Brain and Cognitive Sciences Department
	Organization	Italian Institute of Technology
	Address	Genoa, Italy
	E-mail	vishwanathan.mohan@unige.it
Author	Family name	Morasso
	Particle	
	Given Name	P.
	Suffix	
	Division	Robotics, Brain and Cognitive Sciences Department
	Organization	Italian Institute of Technology
	Address	Genoa, Italy
	Division	DIST
	Organization	University of Genoa
	Address	Genoa, Italy
	E-mail	
Author	Family name	Metta
	Particle	
	Given Name	G.
	Suffix	
	Division	Robotics, Brain and Cognitive Sciences Department
	Organization	Italian Institute of Technology
	Address	Genoa, Italy
	Division	DIST
	Organization	University of Genoa
	Address	Genoa, Italy

	E-mail	
Author	Family name	Sandini
	Particle	
	Given Name	G.
	Suffix	
	Division	Robotics, Brain and Cognitive Sciences Department
	Organization	Italian Institute of Technology
	Address	Genoa, Italy
	Division	DIST
	Organization	University of Genoa
	Address	Genoa, Italy
	E-mail	
Schedule	Received	14 December 2008
	Revised	
	Accepted	21 July 2009
Abstract

This paper addresses the problem of planning the movement of highly redundant humanoid robots based on non-linear attractor dynamics, where the attractor landscape is obtained by combining multiple force fields in different reference systems. The computational process of relaxation in the attractor landscape is similar to coordinating the movements of a puppet by means of attached strings, the strings in our case being the virtual force fields generated by the intended/attended goal and the other task dependent combinations of constraints involved in the execution of the task. Hence the name PMP (Passive Motion Paradigm) was given to the computational model. The method does not require explicit kinematic inversion and the computational mechanism does not crash near kinematic singularities or when the robot is asked to achieve a final pose that is outside its intrinsic workspace: what happens, in this case, is the *gentle degradation* of performance that characterizes humans in the same situations. Further, the measure of inconsistency in the relaxation in such cases can be directly used to trigger higher level reasoning in terms of breaking the goal into a sequence of subgoals directed towards searching and perhaps using tools to realize the otherwise unrealizable goal. The basic PMP model has been further expanded in the present paper by means of (1) a non-linear dynamical timing mechanism that provides terminal attractor properties to the relaxation process and (2) branching units that allow to 'compose' complex PMP-networks to coordinate multiple kinematic chains in a complex structure, including manipulated tools. A preliminary evaluation of the approach has been carried out with the 53 degrees of freedom humanoid robot iCub, with particular reference to trajectory formation and bimanual/whole upper body coordination under the presence of different structural and task specific constraints.

Keywords	Humanoid robots – iCub – Passive motion paradigm – Bimanual coordination – Terminal attractors
Footnotes	The online version of this article (<ExternalRef><RefSource>10.1007/s10514-009-9127-x</RefSource><RefTarget TargetType="DOI" Address="10.1007/s10514-009-9127-x"></ExternalRef>) contains supplementary material, which is available to authorized users.

Auton Robot
DOI 10.1007/s10514-009-9127-x

A biomimetic, force-field based computational model for motion planning and bimanual coordination in humanoid robots

V. Mohan · P. Morasso · G. Metta · G. Sandini

Received: 14 December 2008 / Accepted: 21 July 2009

Abstract This paper addresses the problem of planning the movement of highly redundant humanoid robots based on non-linear attractor dynamics, where the attractor landscape is obtained by combining multiple force fields in different reference systems. The computational process of relaxation in the attractor landscape is similar to coordinating the movements of a puppet by means of attached strings, the strings in our case being the virtual force fields generated by the intended/attended goal and the other task dependent combinations of constraints involved in the execution of the task. Hence the name PMP (Passive Motion Paradigm) was given to the computational model. The method does not require explicit kinematic inversion and the computational mechanism does not crash near kinematic singularities or when the robot is asked to achieve a final pose that is outside its intrinsic workspace: what happens, in this case, is the *gentle degradation* of performance that characterizes humans in the same situations. Further, the measure of inconsistency in the relaxation in such cases can be directly used to trigger higher level reasoning in terms of breaking the goal into a sequence of subgoals directed towards searching and perhaps using tools to realize the otherwise unrealizable goal. The basic PMP model has been further expanded in the present paper by means of (1) a non-linear dynamical timing mechanism that provides terminal attractor properties to the relaxation process and (2) branching units that allow to 'compose' complex PMP-networks to coordinate multiple kinematic chains in a complex structure, including manipulated tools. A preliminary evaluation of the approach has been carried out with the 53 degrees of freedom humanoid robot iCub, with particular reference to trajectory formation and bimanual/whole upper body coordination under the presence of different structural and task specific constraints.

Keywords Humanoid robots · iCub · Passive motion paradigm · Bimanual coordination · Terminal attractors

Electronic supplementary material The online version of this article (http://dx.doi.org/10.1007/s10514-009-9127-x) contains supplementary material, which is available to authorized users.

V. Mohan (✉) · P. Morasso · G. Metta · G. Sandini
Robotics, Brain and Cognitive Sciences Department, Italian Institute of Technology, Genoa, Italy
e-mail: vishwanathan.mohan@unige.it

P. Morasso · G. Metta · G. Sandini
DIST, University of Genoa, Genoa, Italy

1 Introduction

Humanoid robots have a large number of "extra" joints, organized in a humanlike fashion according to several kinematic chains, possibly augmented by the dynamics of manipulated tools. Consider, for example, Cog (Brooks 1997) with 22 DOFs (Degrees of Freedom), DB (Atkeson et al. 2000) with 30 DoFs, Asimo (Hirose and Ogawa 2007) with 34 DoFs, H7 (Nishiwaki et al. 2007) with 35 DoFs, iCub (Metta et al. 2008) with 53 DoFs, to name just a few. When a finger touches a target, the elbow might be up or down and the trunk may be bent forward, backward or sideways. Thus an infinite number of solutions are available to the motor planner/controller. This redundancy is advantageous because it enables a robot to avoid obstacles, joint limits, limb interference and attain more desirable postures, for example when it is not sufficient to simply tap a target because a precise force vector must be applied to the touched object. From a control and learning point of view, however, redundancy also makes it quite complicated to find good movement plans that do not crash when it turns out that the designated target is unreachable or barely reachable.

How do humans decide what to do with their extra joints, and how should humanoid robots control all their joints in order to generate coordinated movement patterns? Moreover, is the selection/coordination of redundant DoFs independent of the spatio-temporal organization of the reaching movements? Early studies of human arm trajectory formation (Morasso 1981; Abend et al. 1982) showed invariant spatio-temporal features, such as a symmetric bell-shaped speed profile, which can be explained in terms of minimization of some measure of *smoothness*, such as jerk (Flash and Hogan 1985) or torque-change (Uno et al. 1989). Later studies suggested that physical or computational force fields can provide constraints for the coordination of multiple joints or motor learning (Mussa Ivaldi et al. 1988; Bizzi et al. 1991; Shadmehr and Mussa-Ivaldi 1994).

Most approaches to motion planning in robotics were derived from the early study of Whitney (1969) named RMRC (Resolved Motion Rate Control), which is based on the real-time inversion of the Jacobian matrix of the kinematic transformation, i.e. the function that links the variation of the joint angle vector dq to the pose dx of the end-effector. Clearly, for redundant kinematic chains RMRC must be modified by using some kind of pseudo-inversion, as the Moore-Penrose inverse that provides a minimum norm solution for dq or other more general pseudo-inversion methods (Liegeosis 1977) that associate an arbitrary cost function to the inversion calculation. In particular, the damped least squares method (DLS, also called the Levenberg-Marquardt method) avoids many of the pseudo-inverse method's problems with singularities (Nakamura and Hanafusa 1986; Wampler 1986; Buss and Kim Jin-Lu 1984?> **?**BusKim1984) but still requires the real-time computation of the matrix inverse. Another method (Extended Jacobian Method: Baillieul 1985; Šoch and Lórencz 2005) extends the usual Jacobian matrix with additional rows that take into account virtual movements in the null space of the kinematic transformation: the extended Jacobian matrix is square and can be inverted in the usual way. Matrix inversion is avoided by the so called Jacobian transpose method (Balestrino et al. 1984; Wolovich and Elliot 1984), which is based on the fact that virtual movements of the end-effector, driven by the transpose Jacobian, tend to reduce the distance of the end-effector from the target in all circumstances. In any case, the classical approaches to robot planning/control work well only inside the workspace and far away from kinematic singularities.

The Passive Motion Paradigm (or PMP: Mussa Ivaldi et al. 1988) is a computational model that addresses the problem of coordinating redundant degrees of freedom by means of a dynamical system approach, similar to the Vector Integration to To Endpoint (VITE model: Bullock and Grossberg 1988). In both cases there is a "difference vector" associated with an attractor dynamics that has a point attractor in the designated target goal. The difference is that the VITE model focuses on the neural signals commanding a pair of agonist-antagonist muscles, whereas the PMP model focuses, at the same time, on the trajectories in the extrinsic and intrinsic spaces. The PMP model exploits the bidirectional mapping between the intrinsic (joints) and extrinsic (end-effector) spaces that characterizes any kinematic chain: the operator that maps incremental motion in the intrinsic space into the corresponding motion in the extrinsic space (i.e. the Jacobian matrix of the kinematic transformation) and the transpose Jacobian that maps efforts in the opposite direction (force at the end-effector into joint torques). The "difference vector" of the VITE model becomes, in the PMP model, a virtual "force field" applied to the end-effector: this field is mapped into the corresponding field in the joint space that determines an elementary motion in agreement with the mechanical "admittance" of the kinematic chain and then, through the forward kinematic operator, a motion of the end-effector in the extrinsic space until the target is reached. This is why the name "Passive Motion" paradigm was used to identify the non-linear dynamic computational mechanism. In fact, it is analogous to the mechanism of coordinating the motion of a wooden marionette by means of attached strings: the motion of the joints is the "passive" consequence of the forces applied to the end effectors. In other words, "passivity" must not be intended in technical sense but as a computational metaphor. The model does not require any cost function to be specified explicitly in order to solve the indeterminacy related to the excess DoFs but it allows to integrate in a task-dependent way, at run-time, internal and external constraints (in the intrinsic and extrinsic spaces, respectively) that automatically solve the coordination problem of the excess DoFs. The computational units of a PMP network operate in different spaces (end-effector space, joint/actuator space, tool space) and locally compute their own reaction to the "source" of planned motion based on their local virtual impedance/admittance. No matrix inversion is necessary and the computational mechanism does not crash near kinematic singularities or when the robot is asked to achieve a final pose that is outside its intrinsic workspace: what happens, in this case, is the *gentle degradation* of performance that characterizes humans in similar situations. Moreover, the remaining error at equilibrium is a valuable information for triggering a higher level of reasoning, such as searching for an alternative plan or making/using an environmental object as a tool.

In this paper, we propose the following two extensions to the basic model necessary for applying it to the complex structure of a humanoid robot:

– Terminal attractor dynamics, by means of a non-linear, dynamic timing mechanism, for allowing the synchronization of kinematic patterns in the extrinsic and intrinsic spaces, bimanual coordination;

Journal ID: 10514, Article ID: 9127, Date: 2009-07-28, Proof No: 1

« AURO 10514 layout: Large v.1.3 reference style: apa file: auro9127.tex (VJ) aid: 9127 doctopic: OriginalPaper class: spr-twocol-v1 v.2009/07/21 Prn:2009/07/28; 12:39 p. 3/17»

– Branching nodes, for structuring PMP-networks in agreement with the body model and the kinematic constraints of a specific task.

The proposed computational model has been evaluated using the 53 degrees of freedom humanoid robot iCub, with particular reference to trajectory formation and bimanual/whole upper body coordination under the presence of different structural and task specific constraints. The model presented in this paper is an evolution of the primitive PMP based computational models like M-Nets and P-Nets (Pagliano et al. 1991), mainly restructured and extended to coordinate complex motion patterns in humanoid robots. The control of the timing of the relaxation process using terminal attractor dynamics endows the generated trajectories with human-like smoothness and is crucial for complex motion patterns such as bimanual coordination, interference avoidance and precise control of the reaching time. The same relaxation process can dynamically coordinate the movements of a single kinematic chain (e.g. upper or lower "limbs"), network of body parts (e.g. left arm–waist–right arm) or networks of external objects kinematically coupled to the body network (e.g. right arm–tool–left arm, as in driving a car or transporting objects using two arms). In this paper, we demonstrate how such custom *PMP-networks* can be assembled at run-time in a flexible, task-oriented manner.

In comparison with a recent paper by Hersch and Billard (2008) that builds upon the VITE model, the proposed model is equally well a "multi-referential dynamical systems" for implementing reaching movements in complex, humanoid robots but does not require any explicit inversion and/or optimisation procedure. Another approach to motion planning, based on non-linear dynamics, has been proposed by Ijspeert et al. (2002) in order to form control policies for discrete movements, such as reaching. The basic idea is to learn attractor landscapes in phase space for canonical dynamical systems with well defined point attractor properties. The approach is very effective for movement imitation, because it approximates the attractor landscape by means of a piecewise-linear regression technique. Also in the PMP model there is a well defined attractor landscape which is derived from the composition of different virtual force fields that have a clear meaning and thus allow the seamless integration of planning with reasoning (Mohan and Morasso 2007). Moreover, the same computational process can be used to perform "mental simulations of an action" in order to detect crucial events that may allow the system to re-plan an action or sequence of actions autonomously, before executing it. A mental simulation need not be a perfect replica of a real movement, but must only be a "sufficiently good" approximation, the approximation level being dictated by the requirements of the task.

From the point of view of neural control of movement, a PMP-network should be considered as a "body schema" or an "internal model" that interfaces higher cognitive levels (reasoning and planning) with lower control levels, related to actuators and body dynamics. It is not a controller in the strict sense and thus it is not concerned with dynamics and actuators. In the demonstration with the iCub robot, whose DoFs are separately controlled by means of standard PIDs loops, the output of a PMP-network provides the reference trajectory for each controller.

We also emphasize that PMP-networks assume that a reasoning mechanism, driven by vision and/or memory, has identified a small set of "keypoints" to reach or track. Consider for example a task in which a bottle must be reached and lifted with coordinated movements of both arms in different conditions: (a) with mirror-like motions of the two arms; (b) with different motions of the two arms if the bottle is displace sideways; (c) with a combination of lifting and rotating; etc. In order to solve the task two complementary problems must be solved: (1) appropriate joint rotation patterns must be generated that capture the overall structure of the action; (2) contact forces must be constrained in order to avoid slipping or other contact-related events. The PMP-network is concerned with the former problem, i.e. to put the bimanual patterns in the right ball-park, irrespective of the superficial properties of the bottle and the fingers. The latter problem, on the contrary, is strongly concerned with the superficial properties and can be designed as a set of reactive modules (reflexes) that modulate the stiffness features of the end-effectors and/or exploit the affordances provided by the roughness/compliance of the object's surface.

The rest of the paper is organized as follows: in Sect. 2.1 we present the basic PMP model. Sections 2.2 and 2.3 describe extensions of the basic PMP network to deal with internal and external constraints imposed on the humanoid robot during the execution of a reaching action. Control over timing of the PMP relaxation using terminal attractor dynamics is described in Sect. 2.4. Combining different PMP relaxations applied to different parts of a complex body (simplest case being of a bimanual coordination task) through formulation of branching nodes is presented in Sect. 2.5. Implementation and evaluation of the computational model on the iCub humanoid platform with focus on bimanual and upper body coordination is described in Sect. 3.1. In Sect. 3.2, using an example of a bimanual transportation task, we describe how the PMP net can further be extended to include dynamics of external objects coupled to the body. We conclude with a discussion on the salient features of the computational model and a brief outline for future work.

Journal ID: 10514, Article ID: 9127, Date: 2009-07-28, Proof No: 1

« AURO 10514 layout: Large v.1.3 reference style: apa file: auro9127.tex (VJ) aid: 9127 doctopic: OriginalPaper class: spr-twocol-v1 v.2009/07/21 Prn:2009/07/28; 12:39 p. 4/17»

2 The computational model

2.1 General formulation

Let x be the vector that identifies the pose of the end-effector of a robot in the extrinsic workspace and q the vector that identifies the configuration of the robot in the intrinsic joint space: $x = f(q)$ is the kinematic transformation that can be expressed, for each time instant, as follows: $\dot{x} = J(q) \cdot \dot{q}$ where $J(q)$ is the Jacobian matrix of the transformation. The motor planner, which expresses in computational terms the PMP (Mussa Ivaldi et al. ?MusIva1998), is defined by the following steps that are also represented graphically by the PMP network of Fig. 1.

(1) *Activate a target-dependent, virtual force field in the extrinsic space*:

$$F = K_{\text{ext}}(x_T - x) \tag{1}$$

where x_T is the target and K_{ext} the virtual stiffness in the extrinsic space. The intensity of this force decreases monotonically as the end-effector approaches the target. The force field described by (1) can be isotropic or anisotropic according to the fact that the eigenvalues of matrix K_{ext} are equal or unequal. The flowlines in the former case are straight lines and are curved in the latter case. More complex curved trajectories can be obtained by adding a rotational component to the convergent force field given by (1).

(2) *Map this field into an equivalent virtual torque field in the intrinsic space according to the principle of virtual works*:

$$T = J^T F \tag{2}$$

Also the intensity of this torque vector decreases as the end-effector incrementally approaches the target.

(3) *Relax the arm configuration in the applied field*:

$$\dot{q} = A_{\text{int}} \cdot T \tag{3}$$

where A_{int} is the virtual admittance matrix in the intrinsic space: the modulation of this matrix affects the relative contributions of the different joints to the overall reaching movement.

(4) *Map the arm movement into the extrinsic workspace*:

$$\dot{x} = J \cdot \dot{q} \tag{4}$$

(5) *Integrate over time until equilibrium*:

$$x(t) = \int_{t_o}^{t} J\dot{q}\,d\tau \tag{5}$$

Integrating (4) over time we obtain a trajectory in the extrinsic space, whose final position corresponds to an equilibrium configuration x_T. By definition, the trajectory of the end-effector is the unique flowline in the force field passing through $x(t_0)$ and converging to x_T. The computational scheme described by (1)–(5) is analogous to the mechanism of coordinating the motion of a wooden marionette by means of attached strings. By simply moving the tip of its hands or legs towards the designated goal using the attached strings, once the tip reaches the intended position, the joint angles automatically reach the intended values. At each time step, the goal induced force field incrementally pulls the end effector towards the target. The computed disturbance forces are incrementally mapped into equivalent torques (this projection is implemented by the transpose Jacobian). The virtual torques now cause an incremental change in joint configuration q in agreement with the admittance matrix A_{int} (that defines the relative contributions of different DoFs to the overall reaching movement). The incremental change in joint space is mapped to the extrinsic space (using the Jacobian matrix) causing a small displacement of the end effector towards the intended target. This process cyclically progresses till the time the algorithm converges to an equilibrium state, which is reached asymptotically in the following conditions:

(a) When the end-effector reaches the target, thus reducing to 0 the force field in the extrinsic space (1);
(b) When the force field in the intrinsic space becomes zero (2), although the force field in the extrinsic space is not null and this can happen in the neighbourhood of kinematic singularities.

Case (a) is the condition of success termination. But also in case (b), in which the target cannot be reached for example because it is outside the workspace, the final configuration has a functional meaning for the motion planner because it encodes geometric information valuable for re-planning (Fig. 1, target B).

Thus, the basic PMP is a robust non-linear dynamic approach to the solution of the inverse kinematic problem that does not require any explicit inversion or optimization task. Redundancy is dealt with by the admittance matrix of the kinematic chain. For example, "freezing" or "unfreezing" a joint can be implemented in a simple way by manipulating the relevant elements of the matrix: moreover, this modulation can be carried out efficiently in real-time, in a task-dependent way.

The basic PMP model also includes two additional elements (Fig. 1): (1) a force field in the intrinsic space for implementing internal constraints, (2) a force field in the extrinsic space for implementing external constraints.

As regards the Jacobian and transpose Jacobian matrices, if an analytic expression is not available or is difficult to

Journal ID: 10514, Article ID: 9127, Date: 2009-07-28, Proof No: 1

« AURO 10514 layout: Large v.1.3 reference style: apa file: auro9127.tex (VJ) aid: 9127 doctopic: OriginalPaper class: spr-twocol-v1 v.2009/07/21 Prn:2009/07/28; 12:39 p. 5/17»

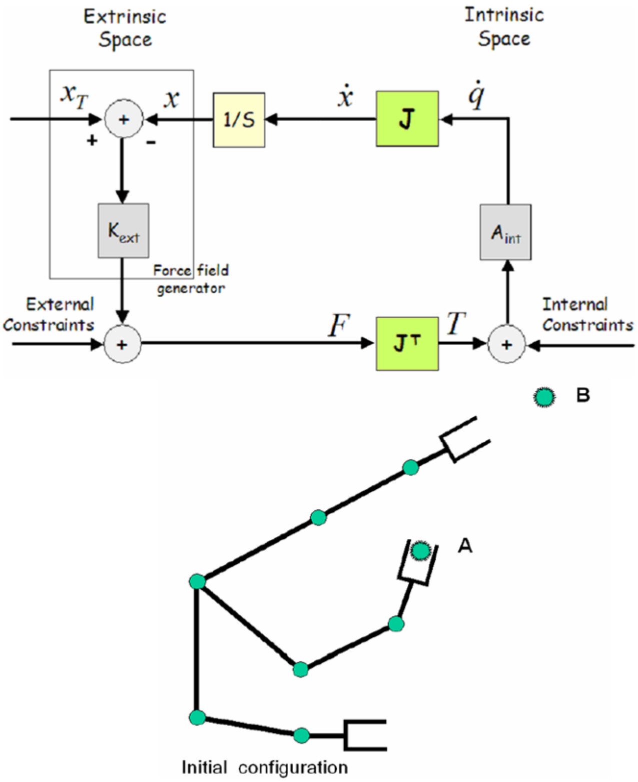


Fig. 1 *Top panel*: Basic computational scheme of the PMP network for a simple kinematic chain. K_{ext} is a virtual stiffness that determines the shape of the attractive force field to the target; "external constraints" are expressed as force fields in the extrinsic space; "internal constraints" are expressed as force fields in the intrinsic space; A_{int} is a virtual admittance that distributes the motion to the different joints. *Bottom panel*: Application of the PMP to a redundant planar robot, starting from a given initial configuration. Target A is inside the workspace and can be reached in infinite possible ways: the actual chosen configuration depends on the "internal constraints" and the admittance matrix. Target B is outside the workspace and the unique equilibrium configuration computed by the network is the one closest to the target

obtain, it is possible to use a neural network representation that is easy to obtain by means of a self-teaching procedure and is computationally efficient for real-time usage (Mohan and Morasso 2007).

2.2 Internal constraints—joint limit avoidance

Typical internal constraints are related to joint limit avoidance, i.e. keeping each DoF inside a given range of motion: $\{q_i^{\min} < q_i < q_i^{\max}, i = 1, n\}$. Such constraints can be implemented by means of a repulsive force field that pushes away the joint angle from limit angles. A suitable profile is an inverted sigmoid (Fig. 2):

$$\begin{cases} \lambda_i = \frac{q_i - q_i^{\min}}{q_i^{\max} - q_i^{\min}} \\ T_i = \alpha \ln \frac{1-\lambda_i}{\lambda_i} \end{cases} \tag{6}$$

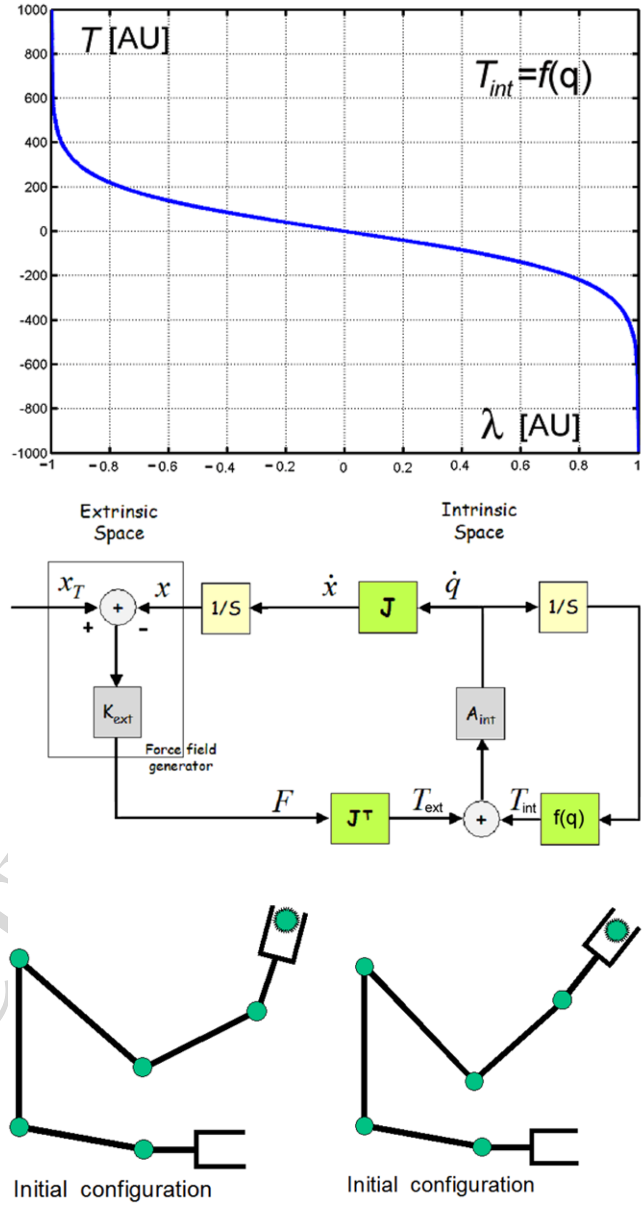


Fig. 2 *Top panel*: normalized profile of the torque for smoothly enforcing joint limit avoidance ($y = \ln \frac{1-x}{x}$). *Middle panel*: PMP network for implementing joint limit avoidance with a suitable intrinsic torque field. *Bottom panel*: two reaching movements to the same target, with the same initial configuration but different joint limits of the wrist

where q_i is the normalized joint rotation angle for the ith joint and α is a scale factor of the intrinsic torque field in relation with the extrinsic force field.

These equations are meant to operate in real-time, generating a torque field that is superimposed on the field mapped from the extrinsic space. In this way, the kinematic redundancy determined by excess DoFs is automatically compensated for by selecting, among the infinite kinematic configurations compatible with the target, the configuration that is farthest from the joint limits.

Journal ID: 10514, Article ID: 9127, Date: 2009-07-28, Proof No: 1

Other robot-dependent constraints can be envisaged, in order to exploit redundancy of the kinematic chain, and can be integrated in the same computational architecture.

2.3 External constraints—torque limit avoidance

External constraints are usually task-dependent. A typical constraint is an obstacle, which can be implemented as a repulsive force field in the extrinsic space, to be added to the attractive force field to the target.

Another important related problem that can be addressed in a similar way is avoiding the torque limits that characterize the actuators: an optimal arm configuration, from the point of view of the actuators, corresponds to a required torque output for each actuator that is as far as possible from the torque limit. This involves a kind of search in the null space of the kinematic transformation because, for a given force vector delivered at the end-effector, the actuator torques depend on the arm configuration via the transpose Jacobian. The solution is given by operating the PMP model, after x has converged to x_T, with the target force F_T applied as an additional input in the extrinsic space. A saturation block, inserted after the mapping from the extrinsic to the intrinsic space (Fig. 3), allows the virtual motion in the null space to settle in a configuration that satisfies the torque limits.

2.4 Control over timing: Terminal attractor dynamics

The basic PMP model is an asympotically stable dynamical system with a point attractor that brings the end-effector to the target if the target is indeed reachable. However, asymptotic stability implies that the equilibrium configuration is reached after an infinite time and does not provide any mechanism to control the speed of approach to equilibrium.

A way to explicitly control time, is to insert in the non-linear dynamics of the PMP model a suitable time-varying gain $\Gamma(t)$ that grows monotonically as x approaches the equilibrium state and diverges to an infinite value in that state. The technique was originally proposed by Zak (1988) for speeding up the access to content addressable memories and then was applied to a number of problems in neural networks. Our purpose, however, is not merely to speed up the operation time of the planner but to allow a control of the reaching time as well, in order to approximate the bell-shaped human speed profile in reaching and allow synchronization e.g. in bimanual coordination or in other complex tasks. In particular, we propose to extend the basic PMP model (Fig. 4) by inserting the time-varying gain $\Gamma(t)$, as a further development of what was proposed by Tsuji et al. (1995) and Morasso et al. (1997). The time-varying gain is defined as follows:

$$\begin{cases} \Gamma(t) = \frac{\dot{\xi}}{(1-\xi)} \\ \xi(t) = 6\cdot(t/\tau)^5 - 15(t/\tau)^4 + 10(t/\tau)^3 \end{cases} \tag{7}$$

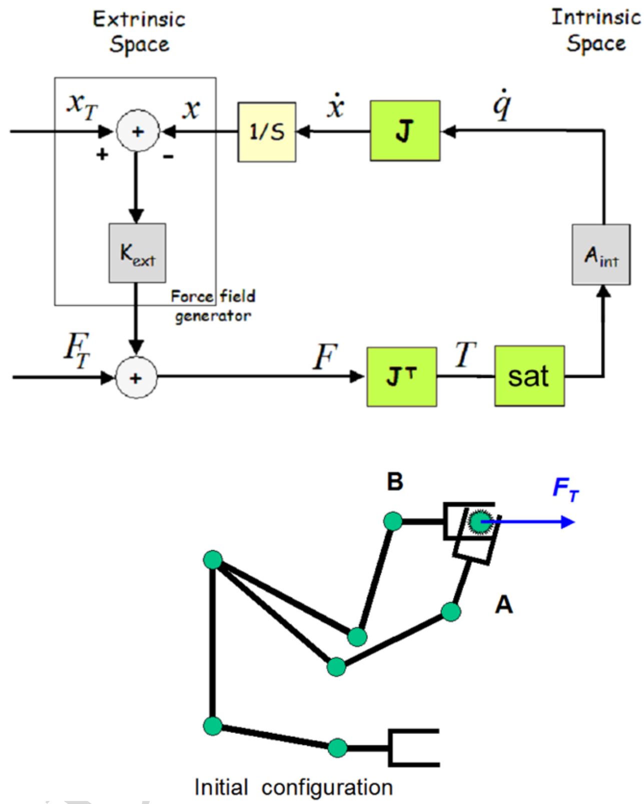


Fig. 3 *Top panel*: PMP network for selecting the best final configuration in relation with a target force vector F_T and the range of torque values for each motor; "sat" is a saturation block that keeps each torque value inside to allowed range. In the *bottom panel* "A" is the final configuration identified by the regular PMP model. "B" is the configuration obtained by allowing the network to settle in the null space of the kinematic transformation (for $x = x_T$) by "saturating" the different actuators to the rated torques. In the example, the wrist motor is much weaker than the elbow and shoulder motors

where $\xi(t)$ is a time-base generator (TBG): a scalar function that smoothly evolves from 0 to 1 with a prescribed duration τ and a symmetric bell-shaped speed profile. A simple choice for the TBG is the minimum jerk polynomial function of (7), but other types of TBGs are also applicable without any loss of generality. In summary, this extension of the basic PMP model in order to allow terminal attractor dynamics simply requires that (4) is substituted by the following one:

$$\dot{x} = \Gamma(t)\cdot J\cdot\dot{q} \tag{4a}$$

In Appendix A we demonstrate that in this way the target is reached after a time equal to τ and with an approximately bell-shaped speed profile. The example of Fig. 4 shows that this simple, non-linear mechanism generates complex coordinated patterns of the different joints without any further explicit computation.

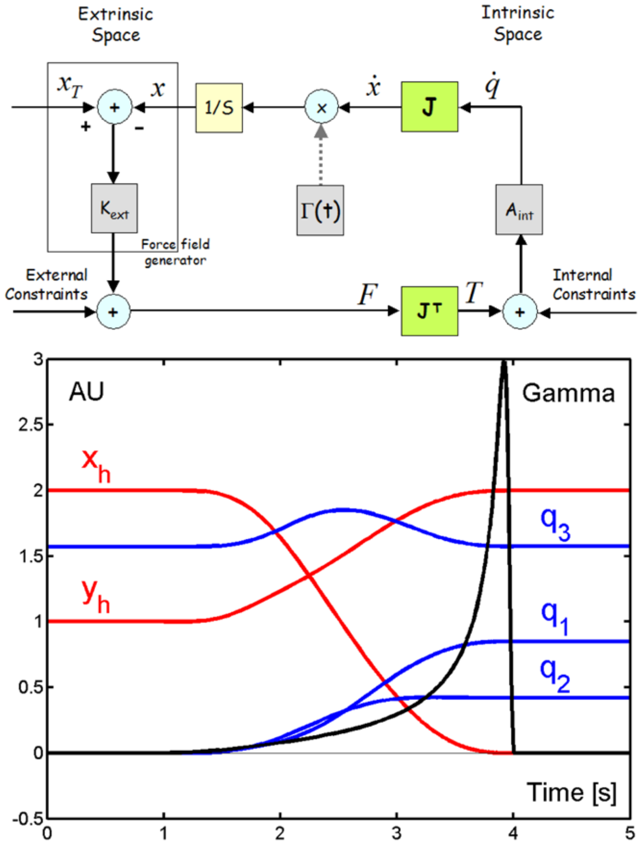


Fig. 4 *Top panel*: PMP network modified with the inclusion of the TBG (Time Base Generator). *Middle panel*: motion patterns generated by the PMP model of a planar 3 DoF manipolandum in the intrinsic space (q) and distal space (x, y)

2.5 Branching nodes, for structuring PMP-networks

In order to apply the PMP model for planning and coordinating the motion of complex humanoid robots, we need to structure and branch the basic PMP-network. This can be achieved by combining a number of PMP networks (one for each limb of the body scheme and/or grasped "tool") by means of two additional nodes, with respect to the scheme of Fig. 1:

– a sum node,
– an assignment node.

The "sum node" allows the force fields applied to the end-effectors of two or more body segments (e.g. the two arms) to be combined in order to propagate the virtual forces to a common body segment (e.g. the trunk). Therefore, the motion of this body segment, far away from the end-effectors, is recruited by the global force fields and modulated by the local admittance matrix. This motion is then reflected back to the impinging segments, by means of an assignment node, thus distributing the movements throughout the overall kinematic structure (Fig. 5). It should be noted that the network can automatically adapt to task features, such as the fact that

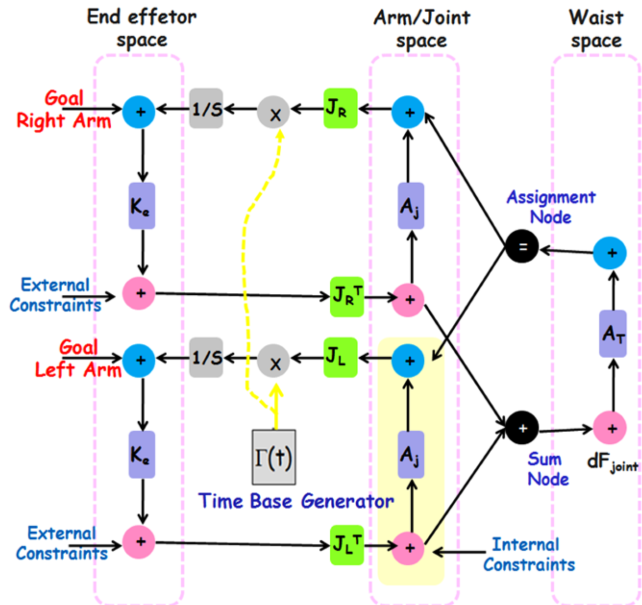


Fig. 5 Composite PMP network with two attractive force fields applied to the right and left arms of a humanoid robot. The "sum node" allows the two force fields to be combined in determining the motion of the trunk. The "assignment node" propagates to the two arms the motion of the trunk. In this way the motion of each arm is influenced by both force fields

the targets are beyond arm's reach. In that case, indeed, we expect an increasing recruitment of the trunk as soon as the arms approach the joint limits. On the contrary, manipulation of targets well inside the workspaces of the arms is likely to induce a very limited involvement of the trunk.

Figure 5 also shows that a single TBG-network can synchronize the action of the reaching movements of the different body segments (left arm-trunk-right arm chain), without any additional coordination process. In this way, for example, the two hands will be able to reach the same target at the same time, whichever the initial distance, just setting $x_{T_1} = x_{T_2}$.

3 PMP-networks for the iCub robotic platform

The iCub is a small humanoid robot of the dimensions of a three and half year old child (Fig. 6) and designed by the RobotCub consortium, a joint collaborative effort of 11 research groups in Europe[1] with an advisory board from

[1] LIRA-Lab, University of Genoa, Italy; ARTS Lab, Scuola Superiore S. Anna, Pisa, Ital; AI Lab, University of Zurich, Switzerland; Dept. of Psychology University of Uppsala, Sweden; Dept. of Biomedical Science, Univ. Ferrara, Italy; Dept. of Computer Science, Univ. of Hertfordshire, UK Computer Vision and Robotics Lab, IST University of Sheffield, UK; Autonomous Systems Lab, Ecole Polytechnique Federal de Lausanne, Switzerland; Telerobot Srl, Genoa Italy; Italian Institute of Technology, Genoa, Italy.

« AURO 10514 layout: Large v.1.3 reference style: apa file: auro9127.tex (VJ) aid: 9127 doctopic: OriginalPaper class: spr-twocol-v1 v.2009/07/21 Prn:2009/07/28; 12:39 p. 8/17»

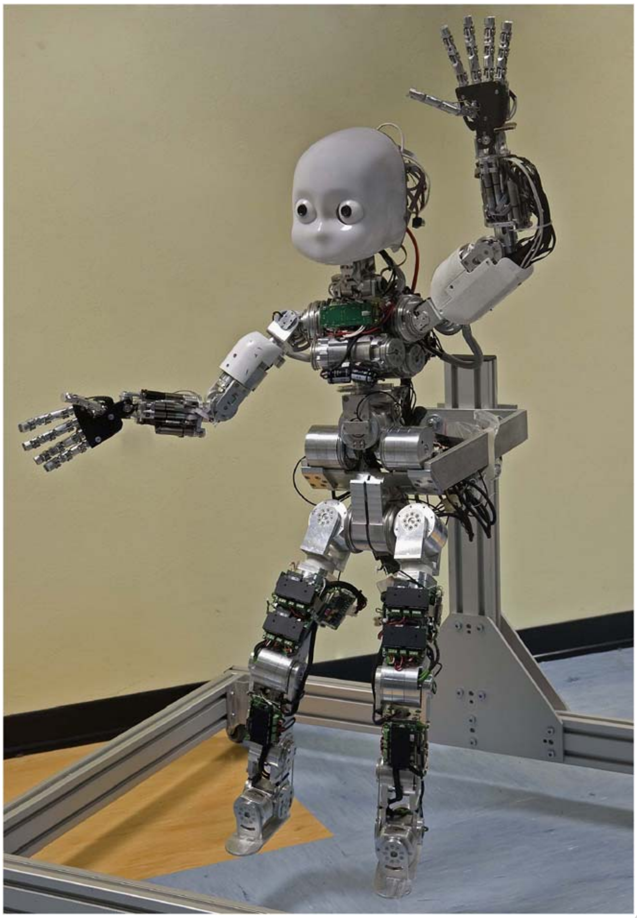


Fig. 6 The 53 DoFs iCub robot

Japan[2] and the USA[3]. The 105 cm tall baby humanoid body is characterized by 53 degrees of freedom: 7 DoF for each arm, 9 for each hand, 6 for the head, 3 for the trunk and spine and 6 for each leg. The current design uses 23 brushless motors in the arms, legs, and the waist joints. The remaining 30 DoFs are controlled by smaller DC motors. The iCub body is also endowed with a range of sensors for measuring forces, torques, joint angles, inertial sensors, tactile sensors, 3 axis gyroscopes, cameras and microphones for visual and auditory information acquisition. Most of the joints are tendon-driven; some are direct-drive, according to the placement of the actuators which is constrained by the shape of the body.

Apart from the interface API that speaks directly to the hardware, the middleware of iCub software architecture is based on YARP (Metta et al. 2006), an open-source framework that supports distributed computation with a specific impetus given to robot control and efficiency. The main goal of YARP is to minimize the effort devoted to infrastructure-level software development by facilitating modularity, support for simultaneous inter-process communication, image processing, as well as a C++ class hierarchy to ease code reuse across different hardware platforms and hence maximize research-level development and collaboration. With special focus being given on manipulation and interaction of the robot with the real world, the iCub is characterized by highly sophisticated hands, flexible oculomotor system and sizable bimanual workspace.

In this study, the computational model was used to coordinate all degrees of freedom involved in the left arm-torso-right arm chain of the baby humanoid (i.e. $7+3+7$ DoFs in total). Specifically, for each arm we deal with the following joints: shoulder pitch (front-back movement when the arm is aligned with gravity), shoulder roll (adduction/abduction movement of the arm), shoulder yaw (yaw movement when the arm principal axis is aligned with gravity), elbow flexion/extension, wrist prono/supination (rotation along arm principal axis), wrist pitch, wrist yaw. While theoretically six DoFs would already allow reaching any point in the workspace with every attainable orientation, in practice, the seventh DoF is necessary to satisfy additional constraints, such as reaching targets in the workspace while avoiding interference with vision. This additional flexibility is very much desired if we have to deal with grasping and the interaction with objects in front of the robot while maintaining sight of the action. It is also worth mentioning that the full range of motion for the shoulder can only be obtained by a double joint mechanism similar to the human clavicle and collar bones. The torso is characterized by three DoFs: torso yaw (with respect to gravity), torso roll (lateral movement) and torso pitch (front back movement). For additional details we refer the interested reader to the RobotCub database (http://www.robotcub.org) as regards the technical description of the body geometry, kinematics, electronics, software architecture and CAD diagrams.

In addition to the YARP middleware, the iCub platform also has a kinematic/dynamic simulator (Tikhanoff et al. 2008). The two software environments are compatible, in the sense that higher-level computational mechanisms, like PMP-networks, can be debugged first by means of the simulator and then applied to the real robot without any change. The simulation phase is also important for verifying if the planned kinematic patterns are compatible with the requirements of the actuators in terms of speed, acceleration, etc. In the following section we show some examples of real and simulated experiments.

[2] MIT Computer Science and Artificial Intelligence Laboratories, Cambridge Mass., USA; University of Minnesota School of Kinesiology, USA.

[3] Communications Research Lab, Japan; Dept. of Mechano-Informatics, Intelligent Informatics Group, University of Tokyo, Japan; ATR Computational Neuroscience Lab, Kyoto, Japan.

Journal ID: 10514, Article ID: 9127, Date: 2009-07-28, Proof No: 1

« AURO 10514 layout: Large v.1.3 reference style: apa file: auro9127.tex (VJ) aid: 9127 doctopic: OriginalPaper class: spr-twocol-v1 v.2009/07/21 Prn:2009/07/28; 12:39 p. 9/17»

3.1 Bimanual coordination

In the paradigm of bimanual coordination, as already noted in the previous section, the basic PMP model must be extended with two additional nodes: a sum node and an assignment node. In complex kinematic structures, characterized by several serial and parallel connections, the sum and assignment nodes can be used to add or assign displacements and forces to different connecting elements of the kinematic chain (in this case the left arm-torso-right arm network). The sum and assignment nodes in general are dual in nature: if an assignment node appears in the kinematic transformation between extrinsic and intrinsic motor spaces, then a sum node appears in the force transformation between the same motor spaces. This is a consequence of conservation of energy that is structurally invariant across the different work units. Figure 5 shows the resulting computational scheme, where A_{trunk} is the virtual admittance matrix of the trunk. We may consider the scheme of Fig. 5 as a composite PMP network, dynamically created for this specific task, reconfiguring the basic PMP networks of the two arms (that were grounded at the shoulder). During the relaxation process, the transpose Jacobian incrementally transform the force fields generated by the goal in each chain into ten virtual torques (7 each for the respective arms and 3 for the waist). The virtual torques incrementally computed for the waist as a result of the force fields experienced by the two arms are summed at the sum node and transformed into three incremental joint rotations at the waist through the admittance matrix (A_{trunk}). The assignment node propagates the resultant incremental displacement computed at the waist back to the computational chain of the two arms. At the same time the incremental displacements at the joints of the each arm is also computed by using the seven virtual joint torques and joint admittance matrices (A_j). The Jacobian matrices now compute the incremental update in the configuration of the body as a result of the incremental displacements at different joints. Hence, in one cycle through the computational chain, the whole upper body has incrementally reconfigured to a new pose towards reaching the respective goals of the two end-effectors. Part of the solution is contributed by the waist, part of it contributed by the degrees of freedom of the two arms, based on their relative admittances. This cycle of propagation of disturbances through the computational chain continues until the whole upper body attains equilibrium (i.e. there are no disturbance forces circulating in the network). This is the final solution of the complete PMP relaxation process.

Figure 7 (panels a–b) shows an example of iCub bimanually reaching a far away target (blue box) using all DoF involved in the 'left arm-trunk-right arm' chain.

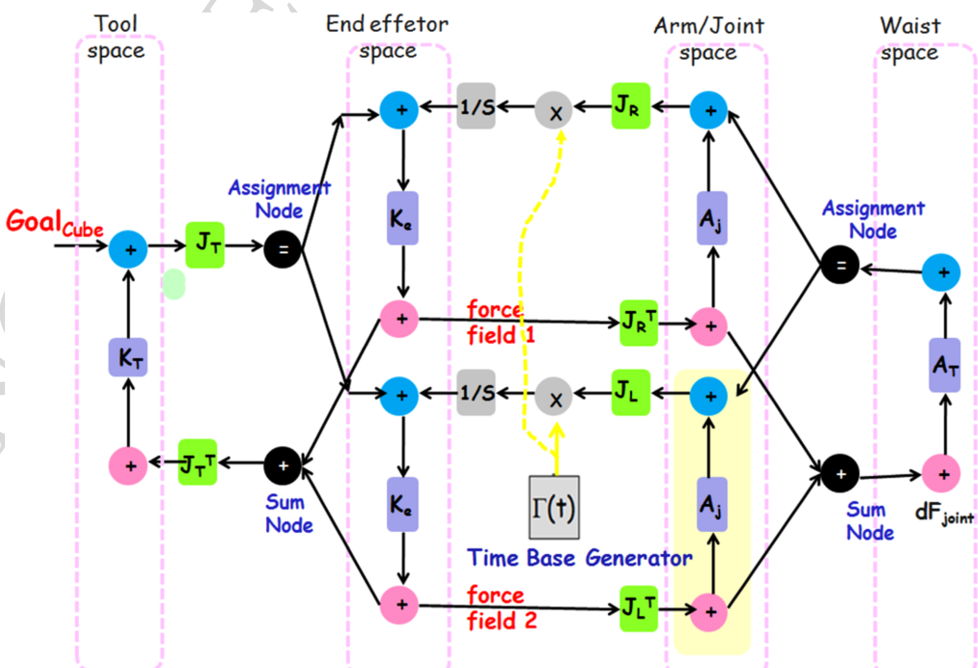
By modulating A_{trunk} we can control the contribution of the trunk DoFs to the overall relaxation of the body in reaction to the two force fields applied to either end-effectors (without affecting the trajectory of the end-effector). If the trunk is very stiff, only the DoFs of the arms contribute to the final solution reached by the system: this is equivalent to "grounding" both shoulders. As seen in panel B (and E–F) the DoFs of the waist are naturally recruited to provide the necessary extension in reach as soon as the arms approach the joint limits. Panel C shows another example of reaching a green cylinder with both arms. For reaching objects placed relatively close to the body, the overall movement is generally distributed among the degrees of freedom of the two arms, the trunk motion being quite minimal. In panels D–F we consider an asymmetric bimanual coordination task. Panel D shows the initial condition with the goal being issued to reach the large cylinder (placed asymmetrically with respect to the robot's body) using both arms; Panels E–F show the solution obtained by the PMP relaxation applied to the upper body in order to achieve the goal. We can observe the contribution of all three DoFs of the torso (coupled with appropriate adjustments in the right arm chain) in order to enable the left arm to cover the additional distance necessary to reach the target (along with the right arm). The timing of the relaxation is controlled using the time base generator. Panels G–I show an example of a stacking task using only the left arm-torso chain. In all experiments, scene analysis and salient point extraction is performed by a visual module; this information is reconstructed in Euclidean space by a 3D reconstruction system (Mohan et al. **?**Mohetal2007) and fed as inputs/goals to the PMP networks. <ref:Mo

3.2 Extending PMP networks to include the dynamics of tools

Figure 8 shows an example of a combined task that steps through two different phases: (1) bimanual reaching of an object, (2) transporting the object held between the two arms to a new target destination. In spite of the fact that humans (almost unconsciously) execute such tasks with noticeable ease, it is worth observing the fact that in this example it is not even straightforward to specify the goal of the task (in computational sense) if we want an artificial agent to do the same. For example, the goal may be verbally described in one way as: 'both arms must move in a coordinated way in order to allow the object coupled in between them to reach its target location in space'. The computational complexity in action generation for this bimanual transportation task stems from the fact that once an object is grasped successfully using the two arms, the task of transporting it poses stringent constraints both on the movement trajectory of the two arms as well as the timing of the motion of both arms. The basic requirement is that both arms must move to the target in such a way that they are always in contact with the object and any unpredictable effect on the object must be

Journal ID: 10514, Article ID: 9127, Date: 2009-07-28, Proof No: 1

Fig. 7 Upper body coordination in iCub using PMP. Panels A–B show iCub bimanually reaching a far away target (*blue box*) using all DoF involved in the 'left arm-trunk-right arm' chain. The solution is generated using the computational model of Fig. 5. As seen in panel B (and E–F) the DoF of the waist are naturally recruited to provide the necessary extension in reach as soon as the arms approach the joint limits. Panel C shows another example of reaching a green cylinder with both arms. Panels D–F show an example of an asymmetric bimanual coordination task. Panels G–I show an example of a stacking task using only the left arm-torso chain

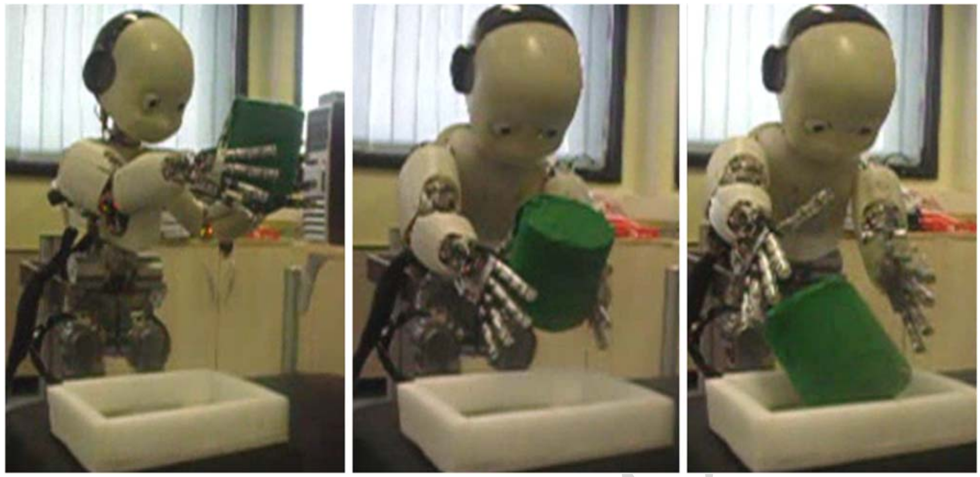


(a)

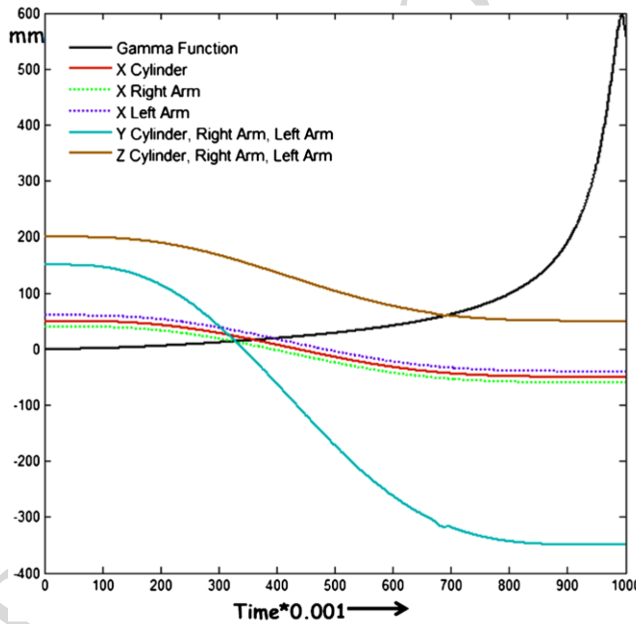
Fig. 8 (**a**) Shows the composite PMP network that coordinates the lifting phase, with a force field applied to the external object (Cylinder) and propagated to the PMP sub-networks that correspond to the right arm, left arm, and trunk. (**b**) Shows a series of snapshots of iCub performing a transportation task. (**c**) Shows the trajectory of the lift phase in the extrinsic space. The TBG function gamma that coordinates the timing of the relaxation is also plotted

« AURO 10514 layout: Large v.1.3 reference style: apa file: auro9127.tex (VJ) aid: 9127 doctopic: OriginalPaper class: spr-twocol-v1 v.2009/07/21 Prn:2009/07/28; 12:39 p. 11/17»

Fig. 8 (*Continued*)



(b)



(c)

compensated by suitable reconfiguration of the body. The planner is further supposed to break down a plan into phases and modify accordingly the PMP network that will carry out each phase independently, selecting the appropriate network parameters. The PMP network that plans the first phase is depicted in Fig. 5, with the two targets being two appropriate points on the object surface. As shown in Fig. 8a, the network for the lift phase is extended to include the manipulated object.

Instead of actively controlling the body that in turn controls the external object and carries it to the goal (which may be extremely complicated to achieve in computational terms considering the number of constraints that must be explicitly accommodated into the controller to achieve this successfully), the computational model shown in Fig. 8a moves exactly in the reverse direction: i.e. the goal pulls the external object that in turn pulls the body (end-effectors) which in turn pulls the intrinsic elements in the body (joints, muscles). Hence the general principle in composing a PMP based forward/inverse model pairs is to always move from the most distal space to the most proximal space (from the goal to the external object, and then to the body).

In this way, the external object in a sense is always kinematically and dynamically coupled with the body. The computational model by itself makes no difference between the representational schema of the motor spaces of the body and the external object. The tool space is represented exactly in the same way as the body, by means of a generalized force and position node, linked vertically by a virtual admittance matrix A_E (characterizing the incremental transformation

Journal ID: 10514, Article ID: 9127, Date: 2009-07-28, Proof No: 1

« AURO 10514 layout: Large v.1.3 reference style: apa file: auro9127.tex (VJ) aid: 9127 doctopic: OriginalPaper class: spr-twocol-v1 v.2009/07/21 Prn:2009/07/28; 12:39 p. 12/17»

from force information to position information in the tool space), and horizontally by the device Jacobian matrix J_D that form the interface between the body and the tool device (and based on the geometry of the task). During the transportation phase, the cube is pulled towards the goal target by one virtual force field; this pull in turn disturbs the end-effectors (both hands) that passively comply to the externally imposed motion; this disturbance then circulates to the proximal space through the Jacobian matrices to derive an incremental change in the joint angles. If the motor commands derived by this process of virtual relaxation are fed to the robot, the robot will reproduce the same motion. The external object can be a simple cylinder, as in Fig. 8b, or a more complicated tool with several new controllable degrees of freedom: for example, two arms linked in parallel to a steering wheel while driving a car. In this case, what changes in the computational model is just the device Jacobian J_E that forms the interface between the body and the external object. In the case of the steering wheel task, J_E maps the transformation between the rotation of the steering wheel and the corresponding differential displacement seen at the end-effectors (hands). Everything else in the computational chain remains exactly the same and behaves accordingly (a mono dimensional steering wheel pattern is mapped to a 6 dimensional end-effector pattern which is mapped to a seven dimensional joint rotation pattern and backwards). Note also that the same time base generator coordinates the joints and the end-effectors position smoothly.

4 Discussion

In this paper, we presented a simple, distributed computational framework for representing and solving a range of difficult coordination problems arising in redundant humanoid platforms, by using a multi-referential, non-linear dynamical approach that exploits the physics of passive virtual motion and the concept of terminal attractor. The virtual force fields representing targets and constraints in different spaces are combined at run-time to yield a net force field that relaxes the internal model to an equilibrium configuration: this solution is the best trade-off among the multiple set of constraints and is computed implicitly by the dynamics of the computational model.

The simplest form of combination of the different force fields is linear superposition. However this is not mandatory. The computational scheme is compatible with methods of shaping the attractor landscapes in terms of basis functions. Future generations of PMP-networks will incorporate mechanisms of this kind. Some of the basis functions can effectively take into account the dynamics of the robot assuming a proper computed torque controller is applicable, thus effectively opening up the possibility of exploiting the robot's dynamics.

The internal model—the force field—stores a whole family of geometrically possible solutions, from which one is implicitly selected based on the nature of the task being executed and attractor dynamics of the system. Further, the proposed architecture is also endowed with nice computational properties like robustness, run-time optimization, fast task adaptation, interference avoidance and local to global computation that make it both biologically plausible and extremely useful in the control of complex robotic bodies.

Robustness

The robustness of the computational machinery stems from several reasons:

(a) No model inversion is needed as the system always operates by means of incremental, well-posed, direct computations.
(b) The dynamical system automatically stays away from singular configurations.
(c) Even if the target is outside the reachable workspace, the robot nevertheless tries to approach the target as much as possible by fully extending the arm to a position that is at a minimum distance from the target. Hence, what we see in such cases is a *gentle degradation* of performance that characterizes humans in the same situations. Although there is no exact solution to the problem, the network "does its best".

Flexibility

Flexibility of the computational machinery is made possible by the following properties:

(a) There is no specific, pre-defined cost function/optimization constraint (minimum torque, minimum jerk, signal dependent noise etc.); hence there is a scope for operating on-line, facilitating run-time co-evolution of plans and the corresponding control processes needed to achieve them.
(b) Multiple constraints can be concurrently imposed in a task-dependent fashion by simply switching on/off different task relevant force field generators.
(c) The same flexibility is also available in the recruitment of different degrees of freedom afforded by a complex body in the performance of a specific task.
(d) Custom PMP-networks containing many kinematic chains and possibly linked with external objects can be composed in a systematic manner, according to the task at hand.

Local to global computing

From the perspective of local to global computing, we can observe that, at each instance of time, every element in the

Journal ID: 10514, Article ID: 9127, Date: 2009-07-28, Proof No: 1

« AURO 10514 layout: Large v.1.3 reference style: apa file: auro9127.tex (VJ) aid: 9127 doctopic: OriginalPaper class: spr-twocol-v1 v.2009/07/21 Prn:2009/07/28; 12:39 p. 13/17»

computational chain of a PMP network makes a local decision regarding its contribution to the overall externally induced pull, based on its own virtual compliance. All such local decisions contribute towards driving the system to a configuration that minimizes its global potential energy. Similar to many connectionist models in the field of artificial neural networks, the mechanism to regularize and exploit redundancy by means of attaining configurations that minimize global potential energy essentially uses only local asynchronous interactions. Analogous to content addressable or auto-associative memories, which reconstruct a stored memory pattern from a partial fragment by filling up all the missing information during the progression to attain an equilibrium state, a plan in the proposed computational model does not need to specify the behavior of all joints and muscles but only requires to specify the desired behaviour of a small number of end-effectors or external tools, because the detailed missing information is automatically filled in by the global attractor dynamics.

Forward/Inverse internal models

An interesting area of research directly related to the model proposed in this paper is the use of forward/inverse internal models, now wide-spread in the field of cognitive science. The existence of neural mechanisms that mimic input/output characteristics and the inverse models of the motor apparatus are supported by several behavioral, neuropsychological and imaging data (Miall and Wolpert 1996; Wolpert and Kawato 1998; Rizzolatti et al. 1997). A forward model or an emulator is a computational mechanism that captures the forward or causal relationship between the inputs and outputs of a system. If we consider the arm as the target system, the forward model predicts the next state (position and velocity), given an initial state and motor command. The inverse model does the opposite: it takes a goal-state as input and produces a sequence of motor commands necessary to achieve it. It is quite easy to observe that a PMP network, in all the different versions considered in this paper, is an integrated forward/inverse model composed of: (1) a forward motor controller that maps tentative trajectories in the intrinsic (joint) space into the corresponding trajectories of the end-effector in the extrinsic workspace and (2) an inverse motor controller that maps desired trajectories of the end-effector into feasible trajectories in the joint space, concurrently taking into account the motion of the end-effector predicted by the forward model. We also note that unlike forward/inverse models using supervised neural networks (e.g. Jordan networks: Jordan 1986), the proposed model using the notion of passive motion paradigm operates by seeking stationary configurations of a non linear dynamical system and is somatotopic in nature.

Mental simulation

The advantages of having forward/inverse models are numerous, ranging from overcoming transductive and transport delays, canceling sensory re-afference, aiding distal supervised learning, and mental simulation of actions, among others (Wolpert et al. **?**Woletal1998). A key element of the proposed architecture is that the same computational model can be used to support mental simulations possibly employed by higher level cognitive layers. The actual delivery of motor commands during movement execution, after consistency of the motor plan has been evaluated by the higher level reasoning process (for example, (i) the goal is reachable directly by the end effector taking into account all the task specific constraints, (ii) the goal is reachable using an available tool, or (iii) the goal is unreachable in which case there is no physical execution of any action at all). In other words, one can reason about reaching without actually reaching and yet use the same neural/computational substrate to do so. This point of view is in agreement with the CODAM concept (Corollary Discharge of Attention Movement, Taylor 2003). The relaxation of the coupled forward/inverse model pair provides a general solution for mentally simulating an action of reaching a target position taking into consideration a range of geometric constraints (range of motion in the joint space, internal and external constraints in the workspace) as well as effort-related constraints (range of torque of the actuators, etc.). If the forward simulation is successful, the movement is executed; otherwise the residual "error" or measure of inconsistency can be used to trigger a higher level of reasoning regarding possible availability of a tool that could be used to get closer to the goal.

Sub-symbolic reasoning: from animals to humanoid robots

In a previous work (Mohan and Morasso 2007), we presented preliminary results of the possible use of the proposed computational architecture coupled with a recurrent neural network to solve the two-sticks problem, a well-known benchmark for animal reasoning, using a simple 5 DoFs arm and 2 cameras.

Experimental studies on animal behaviour generally focus on problems that are of great interest to the cognitive robotics community, mainly attention, categorization, memory, spatial cognition, tool use, problem solving, reasoning, language and social cognition. In addition to revealing the subtle intricacies of the cognitive processes operating inside the animal brain (and mind), these experiments form interesting scenarios for developing-validating computational architectures of cognitive control in robotics. For example, Limongelli et al. (**?**Limetal1995) studied the reasoning powers of chimpanzees to determine if they could extract general

Journal ID: 10514, Article ID: 9127, Date: 2009-07-28, Proof No: 1

« AURO 10514 layout: Large v.1.3 reference style: apa file: auro9127.tex (VJ) aid: 9127 doctopic: OriginalPaper class: spr-twocol-v1 v.2009/07/21 Prn:2009/07/28; 12:39 p. 14/17»

rules in order to obtain a reward by suitable tool use in a scenario consisting of a clear tube, open at both ends, with a food reward inside it that could be pushed out of either end by means of using a stick, which was available to the chimpanzee. The chimps successfully managed to extract the food from the tube by 'reaching and pushing' it with a tool of suitable length. If presented with tools of different lengths during a trial, chimps often chose the most appropriate tool directly and did not employ any trial and error based policy of testing with all the available tools. Tool selectivity is critical for animals because selecting an improper tool incurs costs in terms of time and often results in the potential loss of the food to another predator. As reported by Chappell and Kacelnik (?ChaKac2002) and subsequently confirmed by several others, crows are also very selective in choosing the most appropriate tool suitable for a particular task. In a similar 'pulling the reward out of a tube' task, crows often chose tools that precisely matched the geometry of the tube in which the food was trapped. Several such studies from animal reasoning suggest that a large range of animal species appear to be involved in some form of prospection and reasoning that involves using tools to achieve otherwise unrealizable goals (Boysen and Himes 1999; Emery and Clayton 2004). A computational architecture driving behaviour of cognitive robots must support such virtual executions of goal directed movements (using forward/inverse models) in order to find a feasible course of action, at the same time taking into account a range of bodily, environmental and task specific constraints that are locally present. In case the forward simulation is successful, the movement is executed; otherwise a measure of inconsistency (the geometric information encoded in the virtual simulation of action) can be used to trigger higher level reasoning in order to look for an appropriate tool that suits the task specifications.

A humanoid platform like the iCub, affords the possibility to attempt more complex and challenging scenarios requiring intelligent spatio-temporal coordination of its highly redundant body (sometimes along with 'useful' objects in the environment) in order to realize high level user goals. In particular, we are currently developing a general three-layers computational architecture for robotic reasoning: (1) one layer hosts the extended PMP model for integrating multiple constraints and carry out mental simulations of action sequences or preparing the actual execution; (2) a more abstract computational layer initiates planning at the level of goals, rewards, object actions, situation plan, thus operating in a multi-referential environment (sensorimotor space, action space, work space); (3) a third layer involves active intervention of the robot in the environment in order to enrich its knowledge by active exploration and learning.

Acknowledgements This paper was partly supported by the EU project FP6-003835-GNOSYS and FP7-214668-ITALK.

Appendix A: Terminal attractor dynamics of PMP-networks

Summarizing, a PMP-network is described by the following set of non-linear dynamics equations:

$$\begin{cases} F = K_{\text{ext}}(x_T - x) \\ T = J^T F \\ \dot{q} = A_{\text{int}} T \\ \dot{x} = \Gamma(t) J \dot{q} \\ x(t) = \int_{t_o}^{t} \dot{x}\, dt \end{cases} \tag{A1}$$

In order to demonstrate that in this way the target is reached after a time equal to τ (the duration of the TBG) and with an approximately bell-shaped speed profile, we can substitute the vector (4a) with an equivalent scalar equation in the variable z defined as the running distance from the target along the trajectory generated by the PMP network ($z = 0$ for $x = x_T$):

$$\dot{z} = \Gamma(t) f(z)$$

where $f(z)$ is, by construction, a monotonically increasing function of z which passes through the origin because $x = x_T$ is the point attractor of the dynamical PMP model. Therefore, for $f(z)$ we can formulate the following linear bound:

$$\gamma_{\min} z < f(z) < \gamma_{\max} z \tag{A2}$$

where $\gamma_{\min}, \gamma_{\max}$ are two positive constants. By denoting with γ any value inside the $\gamma_{\min} \rightarrow \gamma_{\max}$ interval, we can write the following equation:

$$\frac{dz}{dt} = -\frac{d\xi/dt}{1-\xi} \gamma z \tag{A3}$$

from which we can eliminate time

$$\frac{dz}{d\xi} = -\frac{\gamma z}{1-\xi} \tag{A4}$$

The solution of this equation is then given by:

$$z(t) = z_0 (1-\xi)^{\gamma} \tag{A5}$$

where z_0 is the initial distance from the target along the trajectory. This means that, as the TBG variable $\xi(t)$ approaches 1, the distance of the end-effector from the target goes down to 0, i.e. the end-effector reaches the target exactly at time $t = \tau$ after movement initiation. Since this applies to both limits of the bound we can write the following bound:

$$z_0 (1-\xi(t))^{\gamma_{\min}} < z(t) < z_0 (1-\xi(t))^{\gamma_{\max}} \tag{A6}$$

Journal ID: 10514, Article ID: 9127, Date: 2009-07-28, Proof No: 1

« AURO 10514 layout: Large v.1.3 reference style: apa file: auro9127.tex (VJ) aid: 9127 doctopic: OriginalPaper class: spr-twocol-v1 v.2009/07/21 Prn:2009/07/28; 12:39 p. 15/17»

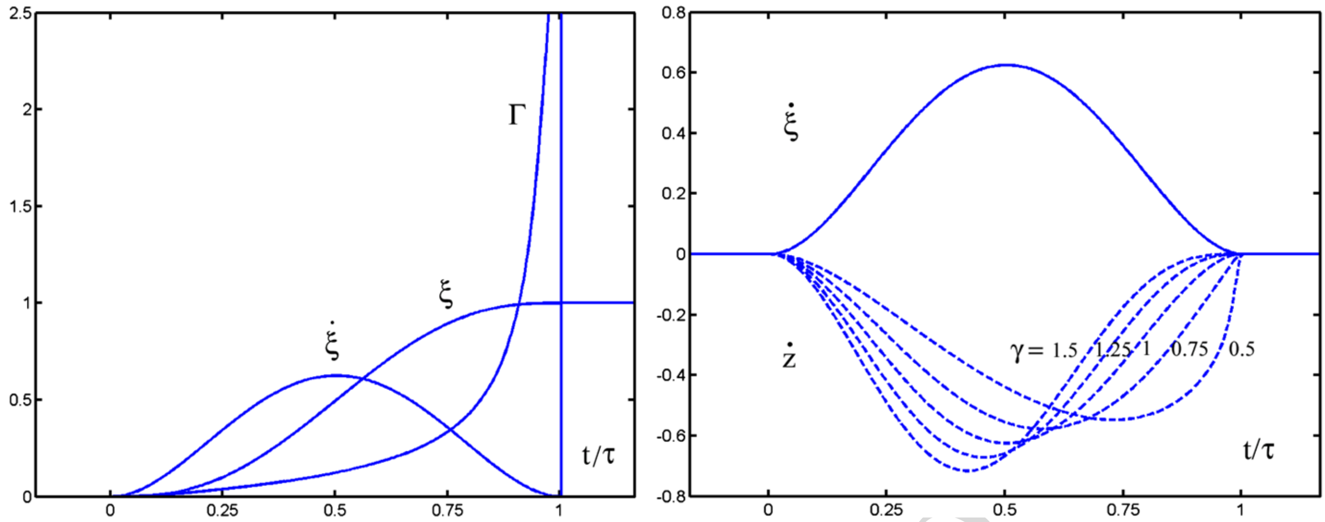


Fig. 9 Time-base generator (TBG) for terminal attractor dynamics—$\Gamma(t)$—obtained from a minimum jerk time function—$\xi(t)$—with assigned duration τ

In any case the terminal attractor $z = 0$ is reached at $t = \tau$. The speed profile may be somehow distorted in relation with a symmetric bell shape (Fig. 9) but the terminal attractor property of the model is maintained for a wide range of values of γ.

References

Abend, W., Bizzi, E., & Morasso, P. (1982). Human arm trajectory formation. *Brain*, *105*, 331–348.

Atkeson, C. G., Hale, J. G., Pollick, F. et al. (2000). Using humanoid robots to study human behavior. *IEEE Intelligent Systems*, *15*, 46–56.

Baillieul, J. (1985). Kinematic programming alternatives for redundant manipulators. In *IEEE international conference on robotics and automation* (pp. 722–728).

Balestrino, A., De Maria, G., & Sciavicco, L. (1984). Robust control of robotic manipulators. In *Proceedings of the 9th IFAC world congress* (Vol. 5, pp. 2435–2440).

Bizzi, E., Mussa Ivaldi, F. A., & Giszter, S. (1991). Computations underlying the execution of movement: A biological perspective. *Science*, *253*, 287–291.

Boysen, S. T., & Himes, G. T. (1999). Current issues and emerging theories in animal cognition. *Annual Reviews of Psychology*, *50*, 683–705.

Brooks, R. A. (1997). The Cog project. *Journal of the Robotics Society of Japan*, *15*, 968–970.

<unc> Brooks, R. A., & Stein, L. A. (1994). Building brains for bodies. *Autonomous Robots*, *1*(1), 7–25.

Bullock, D., & Grossberg, S. (1988). Neural dynamics of planned arm movements: Emergent invariants and speed-accuracy properties. *Psychological Review*, *95*, 49–90.

<unc> Buss, S. R., & Kim, J.-S. (2005). Selectively damped least squares for inverse kinematics. *Journal of Graphics Tools*, *10*(3), 37–49.

<unc> Chiacchio, P., Chiaverini, S., Sciavicco, L., & Siciliano, B. (1991). Closed-loop inverse kinematics schemes for constrained redundant manipulators with task space augmentation and task priority strategy. *International Journal of Robotics Research*, *10*(4), 410–425.

Emery, N. J., & Clayton, N. S. (2004). The mentality of crows: Convergent evolution of intelligence in corvids and apes. *Science*, *306*, 1903–1907.

<unc> Espiau, B., Chaumette, F., & Rives, P. (1992). A new approach to visual servoing in robotics. *IEEE Transactions on Robotics and Automation*, *8*(3).

Flash, T., & Hogan, N. (1985). The coordination of arm movements: an experimentally confirmed mathematical model. *The Journal of Neuroscience*, *5*, 688–703.

Hersch, M., & Billard, A. G. (2008). Reaching with multi-referential dynamical systems. *Autonomous Robots*, *25*(1–2), 71–83.

Hirose, M., & Ogawa, K. (2007). Honda humanoid robots development. *Philosophical Transaction A: Mathematical Physical and Engineering Sciences*, *365*, 11–19.

<unc> Hoffmann, H., Pastor, P., Dae-Hyung, P., & Schaal, S. (2009a). Biologically-inspired dynamical systems for movement generation: Automatic real-time goal adaptation and obstacle avoidance. In *ICRA* 2009.

<unc> Hoffmann, H., Pastor, P., Asfour, T., & Schaal, S. (2009b). Learning and generalization of motor skills by learning from demonstration. In *ICRA* 2009.

Ijspeert, A. J., Nakanishi, J., & Schaal, S. (2002). Movement imitation with nonlinear dynamical systems in humanoid robots. In *Proceed IEEE ICRA2002* (pp. 1398–1403).

Jordan, M. I. (1986). Attractor dynamics and parallelism in a connectionist sequential machine. In *Proc. eighth ann conf cognitive science society* (pp. 531–546). Hillsdale: Erlbaum.

<unc> Khatib, O. (1987). A unified approach for motion and force control of robot manipulators: The operational space formulation. *IEEE Journal of Robotics and Automation*, *3*(1), 43–53.

Liegeosis, A. (1977). Automatic supervisory control of the configuration and behavior of multibody mechanisms. *IEEE Transactions on Systems, Man, and Cybernetics*, *7*, 868–871.

Metta, G., Fitzpatrick, P., & Natale, L. (2006). YARP: Yet another robot platform. *International Journal of Advanced Robotics Systems*, *3*, 43–48.

Metta, G., Sandini, G., Vernon, D., Natale, L., & Nori, F. (2008). The iCub humanoid robot: An open platform for research in embodied cognition. In *PerMIS: Performance metrics for intelligent systems workshop*, Aug 19–21, 2008. Washington DC: USA.

Miall, R. C., & Wolpert, D. M. (1996). Forward models for physiological motor control. *Neural Networks*, *9*, 1265–1279.

Journal ID: 10514, Article ID: 9127, Date: 2009-07-28, Proof No: 1

« **AURO 10514** layout: Large v.1.3 reference style: apa file: auro9127.tex (VJ) aid: 9127 doctopic: OriginalPaper class: spr-twocol-v1 v.2009/07/21 Prn:2009/07/28; 12:39 p. 16/17»

Mohan, V., & Morasso, P. (2007). Towards reasoning and coordinating action in the mental space. *International Journal of Neural Systems*, *17*(4), 1–13.

Morasso, P. (1981). Spatial control of arm movements. *Experimental Brain Research*, *42*, 223–227.

Morasso, P., Sanguineti, V., & Spada, G. (1997). A computational theory of targeting movements based on force fields and topology representing networks. *Neurocomputing*, *15*, 414–434.

Mussa Ivaldi, F. A., Morasso, P., & Zaccaria, R. (1988). Kinematic networks. A distributed model for representing and regularizing motor redundancy. *Biological Cybernetics*, *60*, 1–16.

Nakamura, Y., & Hanafusa, H. (1986). Inverse kinematics solutions with singularity robustness for robot manipulator control. *Journal of Dynamic Systems, Measurement, and Control*, *108*, 163–171.

<unc> Natale, L., Orabona, F., Metta, G., & Sandini, G. (2007). Sensorimotor coordination in a "baby" robot: Learning about objects through grasping. *Progress in Brain Research*, *164*, 403–424.

Nishiwaki, K., Kuffner, J., Kagami, S., Inaba, M., & Inoue, H. (2007). The experimental humanoid robot H7: A research platform for autonomous behaviour. *Philosophical Transaction A: Mathematical Physical and Engineering Sciences*, *365*, 79–107.

Pagliano, S., Sanguineti, V., & Morasso, P. (1991). A neural framework for robot motor planning. In *IEE/RSJ international workshop on intelligent robots and systems IROS '91*.

Rizzolatti, G., Fadiga, L., Fogassi, L., & Gallese, G. (1997). The space around us. *Science*, 190–191.

<unc> Shadmehr, R., & Krakauer, J. W. (2008). A computational neuroanatomy for motor control. *Experimental Brain Research*, *185*, 359–381.

Shadmehr, R., & Mussa-Ivaldi, F. A. (1994). Adaptive representation of dynamics during learning of a motor task. *The Journal of Neuroscience*, *14*, 3208–3224.

Šoch, M., & Lórencz, R. (2005). Solving inverse kinematics—a new approach to the extended Jacobian technique. *Acta Polytechnica*, *45*, 21–26.

Taylor, J. G. (2003). The CODAM model and deficits of consciousness. In *Lecture notes in computer science* (Vol. 2774/2003). Berlin/Heidelberg: Springer.

Tikhanoff, V., Cangelosi, A., Fitzpatrick, P., Metta, G., Natale, L., & Nori, F. (2008). An open-source simulator for cognitive robotics research. *Cogprints*, article 6238.

Tsuji, T., Morasso, P., Shigehashi, K., & Kaneko, M. (1995). Motion planning for manipulators using artificial potential field approach that can adjust convergence time of generated arm trajectory. *Journal Robotics Society of Japan*, *13*, 285–290.

Uno, Y., Kawato, M., & Suzuki, R. (1989). Formation and control of optimal trajectory in human multijoint arm movement: minimum torque-change model. *Biological Cybernetics*, *61*, 89–101.

<unc> Visalberghi, E., & Tomasello, M. (1997). Primate causal understanding in the physical and in the social domains. *Behavioral Processes*, *42*, 189–203.

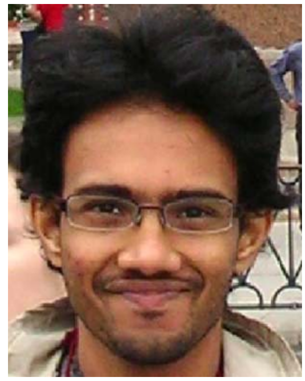
Wampler, C. W. (1986). Manipulator inverse kinematic solutions based on vector formulations and damped least squares methods. *IEEE Transaction on Systems, Man, and Cybernetics*, *16*, 93–101.

Whitney, D. E. (1969). Resolved motion rate control of manipulators and human prosthesis. *IEEE Transactions on Man-Machine Systems*, *MMS-10*, 47–53.

Wolovich, W. A., & Elliot, H. (1984). A computational technique for inverse kinematics. In *Proceedings of the 23rd IEEE conf. on decision and control* (pp. 1359–1363).

Wolpert, D. M., & Kawato, M. (1998). Multiple paired forward and inverse models for motor control. *Neural Networks*, *11*, 1317–1329.

Zak, M. (1988). Terminal attractors for addressable memory in neural networks. *Physics Letters*, *133*, 218–222.



V. Mohan is a post doctoral researcher at the Robotics, Brain and Cognitive sciences department of the Italian Institute of Technology (IIT), a foundation established jointly by the Italian Ministry of Education, Universities and Research and the Ministry of Economy and Finance. He holds an M. Tech in Microelectronics and VLSI design form the Indian Institute of Technology Madras (2003) and a PhD (2009) in Humanoid robotics form the Italian Institute of Technology and University of Genoa. During the course of his doctoral studies, he has worked extensively on the development of the reasoning and action generation system of the GNOSYS robot under the framework of the European project GNOSYS. As a Postdoc at RBCS, in addition to trying to further extend the GNOSYS architecture to a world where many atomic cognitive agents coevolve, learn and cooperate, he is also involved in the EU funded iTalk project trying to understand the neural foundations of 'Language', the supreme faculty that makes us coevolve, cooperate and communicate. From a long term perspective, he is curious to understand the fundamental principles that underlie the transformations between three major forms of causality in nature: the Physical, the Neural and the Mental.

P. Morasso is full professor of Anthropomorphic Robotics and head of the Biomedical Engineering program at the University of Genova. He has a long-standing collaboration with Emilio Bizzi at the Brain and Cognitive Sciences Dept. of MIT. His current interests include, neural control of movement, motor learning, haptic perception, robot therapy. He is author of about 350 papers of which 38 indexed on Medline.

G. Metta is a senior scientist at the IIT and assistant professor at the University of Genoa where he teaches courses on anthropomorphic robotics and intelligent systems for the bioengineering curricula. He holds a MS with honors (in 1994) and PhD (in 2000) in electronic engineering both from the University of Genoa. From 2001 to 2002 he was postdoctoral associate at the MIT AI-Lab where he worked on various humanoid robotic platforms. He is assistant professor at the University of Genoa since 2005 and with IIT since 2006. His research activities are in the fields of biologically motivated and humanoid robotics and in particular in developing lifelong developing artificial systems that show some of the abilities of natural systems. His research developed in collaboration with leading European and international scientists from different disciplines like neuroscience, psychology, and robotics. Giorgio Metta is the author of approximately 90

Journal ID: 10514, Article ID: 9127, Date: 2009-07-28, Proof No: 1

publications and has been working as research scientist and co-PI in several international and national funded projects.

G. Sandini is director of Research at the Italian Institute of Technology and full professor of bioengineering at the University of Genoa. His research activities are in the fields of Biological and Artificial Vision, Computational and Cognitive Neuroscience and Robotics with the objective of understanding the neural mechanisms of human sensory-motor coordination and cognitive development from a biological and an artificial perspective. A distinctive aspect of his research has been the multidisciplinarity of the approach expressed through national and international collaborations with neuroscientists and developmental psychologists. Giulio Sandini is author of more than 300 publications and five international patents. He has been coordinating international collaborative projects since 1984 and has served in the evaluation committees of national and international research funding agencies, research centers and international journals.

Journal ID: 10514, Article ID: 9127, Date: 2009-07-28, Proof No: 1